CLAIMS

We claim:

1. A method to transfer data from a sending application to a receiving application in a computer environment, the method comprising the steps of:

if the receiving application posts a receive buffer exceeding a threshold size when posting a send for a pre-selected number of initial data blocks:

transferring subsequent data having sizes greater than the threshold size using direct memory access read operations:

detecting if the receiving application posts the receive buffer prior to posting the send;

if the receiving application posts the receive buffer prior to posting the send:

sending data and a RDMA receive advertisement in a message if the receiving application posts a send buffer having a size below the threshold size and one of data and RDMA Read information has not been received.

2. The method of claim 1 wherein the step of detecting if the receiving application posts the receive buffer prior to posting the send comprises the steps of:

determining if the receiving application posts a large receive buffer;

determining if the sending application does a send causing the receive posted by the receiving application to complete; and

determining if the receiving application does a small send.

3. The method of claim 1 further comprising the steps of :

detecting if the receiving application posts the receive buffer after posting the send;

if the receiving application posts the receive buffer after posting the send:

copying data to a send buffer having sufficient space at a beginning of the send buffer to put a receive advertisement in a header when a small send happens;

starting a timer;

putting the receive advertisement in the header if the receiving application posts a receive buffer exceeding the threshold size before the timer expires; and

sending the message.

4. The method of claim 3 wherein the step of detecting if the receiving application posts the receive buffer after posting the send comprises the steps of:

determining if the receiving application posts a large receive buffer;

determining if the receiving application performs a small send; and

determining if the sending application does a send causing the receive posted by receiving application to complete.

5. The method of claim 3 further comprising the step of putting a standard header onto the send data if the receiving application does not post a receive buffer exceeding the threshold size before the timer expires.

6. The method of claim 1 wherein the computer environment has at least one system area network.

7. A method to minimize kernel calls in an operating system on a per-socket basis during send and receive operations, the method comprising the steps of:

detecting when an event has occurred that triggers a condition for a select request;

setting a flag to identify the condition;

checking a count of outstanding select requests associated with the condition;

if the count of outstanding select requests is greater than zero:

determining if there is at least one select request associated with the condition;

satisfying each select request associated with condition; and

decrementing the count of outstanding requests for each select request associated with the condition.

8. The method of claim 7 wherein the step of determining if there is at least one select request associated with the condition comprises the step of making a kernel call to take a lock, check under the lock for select requests associated with the condition, and release the lock.

9. The method of claim 7 wherein the step of detecting when an event has occurred that triggers a condition for a select request comprises the step of detecting when one of data is available to be received and a window is opening for a send operation.

10. The method of claim 9 wherein the step of detecting when one of data is available to be received and a window is opening for a send operation comprises the step of detecting when one of normal data is available to be received, out-of-band data is available to be received, and a window is opening for a send operation.

11. The method of claim 7 further comprising the step of incrementing the count in response to an application submitting a select request for the condition.

12. The method of claim 11 further comprising the steps of:

determining if the condition has been satisfied;

if the condition has been satisfied:

decrementing the count; and

satisfying the select request submitted by the application.

13. The method of claim 12 further comprising the step of informing the application to wait for the condition if the condition has not been satisfied.

14. The method of claim 12 wherein the step of determining if the condition is satisfied comprises the steps of:

taking a lock that protects a select request association;

checking under the lock to determine if the condition is satisfied; and

releasing the lock.

15. A computer-readable medium having computer-executable instructions to transfer data from a sending application to a receiving application in a computer environment using direct memory access read operations, the computer-executable instructions performing the steps comprising:

detecting if the receiving application posts the receive buffer prior to posting the send;

if the receiving application posts a receive buffer prior to posting a send:

sending data and a RDMA receive advertisement in a message if the receiving application posts a send buffer having a size below the threshold size and one of data and RDMA Read information has not been received.

16. The computer-readable medium of claim 15 wherein the step of detecting if the receiving application posts the receive buffer prior to posting the send comprises the steps of:

determining if the receiving application posts a large receive buffer;

determining if the sending application does a send causing the receive posted by the receiving application to complete; and

determining if the receiving application does a small send.

17. The computer-readable medium of claim 15 having further computer-executable instructions for performing the steps comprising:

detecting if the receiving application posts the receive buffer after posting the send;

if the receiving application posts the receive buffer after posting the send:

copying data to a send buffer having sufficient space at a beginning of the send buffer to put a receive advertisement in a header when a small send happens;

starting a timer;

putting the receive advertisement in the header if the receiving application posts a receive buffer exceeding the threshold size before the timer expires; and

sending the message.

18. The computer-readable medium of claim 17 wherein the step of detecting if the receiving application posts the receive buffer after posting the send comprises the steps of:

determining if the receiving application posts a large receive buffer;

determining if the receiving application performs a small send; and

determining if the sending application does a send causing the receive posted by receiving application to complete.

19. The computer-readable medium of claim 17 having further computer-executable instructions for performing the step comprising putting a standard header onto the send data if the receiving application does not post a receive buffer exceeding the threshold size before the timer expires.

20. The computer-readable medium of claim 15 wherein the computer environment has at least one system area network.

21. A computer-readable medium having computer-executable instructions for minimizing kernel calls in an operating system on a per-socket basis during send and receive operations, the computer-executable instructions performing the steps comprising:

detecting when an event has occurred that triggers a condition for a select request;

setting a flag to identify the condition;

checking a count of outstanding select requests associated with the condition;

if the count of outstanding select requests is greater than zero:

determining if there is at least one select request associated with the condition;

satisfying each select request associated with condition; and

decrementing the count of outstanding requests for each select request associated with the condition.

22. The computer-readable medium of claim 21 wherein the step of determining if there is at least one select request associated with the condition comprises the step of making a kernel call to take a lock, check under the lock for select requests associated with the condition, and release the lock.

23. The computer-readable medium of claim 22 wherein the step of detecting when an event has occurred that triggers a condition for a select request comprises the step of detecting when one of data is available to be received and a window is opening for a send operation.

24. The computer-readable medium of claim 23 wherein the step of detecting when one of data is available to be received and a window is opening for a send operation comprises the step of detecting when one of normal data is available to be received, out-of-band data is available to be received, and a window is opening for a send operation.

25. The computer-readable medium of claim 21 having further computer-executable instructions for performing the step comprising incrementing the count in response to an application submitting a select request for the condition.

26. The computer-readable medium of claim 25 having further computer-executable instructions for performing the step comprising:

determining if the condition has been satisfied;

if the condition has been satisfied:

decrementing the count; and

satisfying the select request submitted by the application.

27. The computer-readable medium of claim 26 having further computer-executable instructions for performing the step comprising informing the application to wait for the condition if the condition has not been satisfied.

28. The computer-readable medium of claim 26 wherein the step of determining if the condition is satisfied comprises the steps of:

taking a lock that protects a select request association;

checking under the lock to determine if the condition is satisfied; and

releasing the lock.